LOW-COST HIGH-SPEED TECHNIQUES FOR REAL-TIME SIMULATION OF POWER ELECTRONIC SYSTEMS

ONR Grant #N00014-04-1-0373

FINAL REPORT

R.E. Crosbie, J.J. Zenor, R. Bednar, D. Word and N.G. Hingorani

June 2007

McLeod Institute of Simulation Sciences
California State University, Chico

20080715 178

DEFENSE TECHNICAL INFORMATION CENTER

Information for the Defense Community

Month		Day		Year			
0	9	0	3	2	0	0	8

DTIC® has determined on 09 03 2008 that this Technical Document has the Distribution Statement checked below. The current distribution for this document can be found in the DTIC® Technical Report Database.

☒ **DISTRIBUTION STATEMENT A.** Approved for public release; distribution is unlimited.

☐ **© COPYRIGHTED.** U.S. Government or Federal Rights License. All other rights and uses except those permitted by copyright law are reserved by the copyright owner.

☐ **DISTRIBUTION STATEMENT B.** Distribution authorized to U.S. Government agencies only. Other requests for this document shall be referred to controlling office.

☐ **DISTRIBUTION STATEMENT C.** Distribution authorized to U.S. Government Agencies and their contractors. Other requests for this document shall be referred to controlling office.

☐ **DISTRIBUTION STATEMENT D.** Distribution authorized to the Department of Defense and U.S. DoD contractors only. Other requests shall be referred to controlling office.

☐ **DISTRIBUTION STATEMENT E.** Distribution authorized to DoD Components only. Other requests shall be referred to controlling office.

☐ **DISTRIBUTION STATEMENT F.** Further dissemination only as directed by controlling office or higher DoD authority.

Distribution Statement F is also used when a document does not contain a distribution statement and no distribution statement can be determined.

☐ **DISTRIBUTION STATEMENT X.** Distribution authorized to U.S. Government Agencies and private individuals or enterprises eligible to obtain export-controlled technical data in accordance with DoDD 5230.25.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
06-30-2008	Final Report	03/01/04 - 12/31/07

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Low-Cost High-Speed Techniques for Real-Time Simulation of Power Electronic Systems	N00014-04-0373
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
R.E. Crosbie J.J. Zenor R. Bednar D. Word N.G. Hingorani	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
The CSU, Chico Research Foundation Building 25 Main Street Chico, CA 95929-0870	60296
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	**10. SPONSOR/MONITOR'S ACRONYM(S)**
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

13. SUPPLEMENTARY NOTES

14. ABSTRACT

See attached final report.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Roy Crosbie
				11	19b. TELEPHONE NUMBER *(Include area code)* (530) 898-4489

LOW-COST HIGH-SPEED TECHNIQUES FOR REAL-TIME SIMULATION OF POWER ELECTRONIC SYSTEMS

ONR Grant #N00014-04-1-0373
Final Report

TABLE OF CONTENTS

LOW-COST HIGH-SPEED TECHNIQUES FOR REAL-TIME SIMULATION OF POWER ELECTRONIC SYSTEMS

ONR Grant #N00014-04-1-0373
Final Report

1.0 BACKGROUND AND RESEARCH AIMS

1.1 Introduction.

A research team at the McLeod Institute of Simulation Sciences (MISS) at California State University, Chico (CSUC) has been developing high-speed, real-time simulations of power electronic systems since 1999. At that time, real-time simulators were available mainly for large-scale power utility applications and offered minimum frame times in the region of 50 μS. These simulators were customized to specific applications and were expensive (typically from hundreds of thousands to millions of dollars). With the rapid growth of power electronic applications along with the increasing use of higher-frequency, pulse-width modulation (PWM) controllers a need arose for higher-speed, but lower cost, real-time simulators. This was the motivation for the original research effort at Chico. The principal factors offering the potential for success were the availability of alternative computing architectures capable of achieving fast real-time operation, and the recognition that the algorithms commonly used for power system simulation, including those used in real-time simulation, had changed very little over almost four decades. It was felt that a fresh approach to the choice of numerical techniques offered the potential for significant improvement.

Real-time simulation, in which simulated time is exactly equal to real time, is required whenever there is a need to interface the simulation to real hardware, to embedded software running at normal speed, or to a human operator. Hardware-in-the-loop (HIL) simulation, which is interpreted here to include embedded controllers in the loop, provides a convenient, safe and economical test environment for hardware components and subsystems. In the power electronics field, real controllers may be interfaced to a real-time simulator of the actual power system for test and evaluation. The controller will normally incorporate one or more embedded processors that execute the control algorithm. A real-time simulator might also be connected to real electrical machines through suitable interfaces or driven by real a.c. waveforms representing the 3-phase a.c. input to a system.

Earlier research established techniques for low-cost, high-speed, real-time (HSRT) simulation using small arrays of digital signal processors [1][2]. The initial aim was to use this approach to achieve frame times of no more than 10 μS for typical power electronic applications. This goal was met and frame times were subsequently reduced to less than 5 μS using small arrays of digital signal processors connected to the PCI bus of conventional desktop computers. This approach allowed the time-critical real-time simulation code to be executed in parallel on the four available digital signal processor

chips with no penalty of lost cycles caused by operating system interrupts. This is in contrast to the commonly used approach of performing real-time simulations on top of a real-time Linux operating system which is generally agreed to put a lower limit of the order of 10 μS on achievable frame times. There is, however, a cost that must be paid to achieve superior performance in the more labor-intensive model development approach which is required to ensure that the model is efficiently partitioned between processors, that all redundant cycles are eliminated, and that I/O is performed efficiently.

1.2 Status of Research at Start of Period of Performance

Earlier research carried out under ONR Award #N00014-01-1-0394 made progress in a number of areas [3].

First was the development, in a number of stages, of a benchmark simulation used to compare the performance of different approaches. The benchmark consists of a 6-pulse, back-to-back system with filters and PWM controllers. The math model has 23 differential equations, switching logic and the sine/triangle wave comparisons and PID control algorithms for the two PWM controllers. To improve flexibility the simulation was divided into separate coupled modules. This does, however, tend to increase computation time. As the research proceeded newer, more powerful processors became available. Ultimately, using Analog Devices TS201 processors, a minimum frame time of 2.02 μS was achieved. A 12-pulse version of the model involving 39 differential equations achieved a minimum frame time of 4.5 μS.

Second, these simulations were integrated with the Virtual Test Bed (VTB) user interface [4][5] and its Virtual Extension Engine (VXE) graphics utility. This provides a powerful means of establishing model and simulation parameters, of controlling the execution of the simulation, and of displaying the output waveforms.

The research also featured extensive numerical analysis in support of the choice of numerical integration algorithm and stability studies [6]. This led to the choice of state-transition methods for the solution of the model differential equations. It also provided a valuable method of model verification by checking the theoretical predictions of stable regions with the behavior of the simulation. Discrepancies between the two are normally a reliable indication of programming errors. A further study involved the use of frequency domain techniques to aid step-size selection as an alternative to the more common time-domain methods. There was also some investigation of hardware-in-the-loop simulation. This was based on the use of an emulated digital controller implemented on a separate PC platform communicating with the real-time simulator via digital and analog interface

One of the challenges of implementing high-speed real-time simulations using digital signal processor arrays is the labor intensive program development involving manual derivation of the describing differential equations, their conversion to difference equations, and careful hand coding of the simulation. Consequently the use of automatic methods of software development was a major thrust of the earlier research. Several approaches to achieving automation of at least some stages of the model development

process were investigated. Some Matlab routines were developed to automate the process of combining the model differential equations with the numerical integration algorithm to produce the difference equations that were directly coded in the C program.

1.3 Goals of the Current Project

The goals of this project were to continue the effort to reduce minimum frame times, to investigate the use of different types of processor, to continue the effort to automate the model development process, and to develop processes for the implementation of simulations of complete power systems using high-speed, real-time techniques.

2.0 UTILIZATION OF HIGH-SPEED REAL-TIME TECHNIQUES

2.1 Multi-Rate Simulation

Typical applications in power electronics involve the simulation of large systems made up of combinations of a range of electronic, electromechanical, mechanical, thermal and other components and effects covering a wide dynamic range. The components that demand the use of HSRT simulation typically form only a small part of the total system. It is neither feasible nor advisable to try to simulate the entire system using the shortest required frame time. For real-time simulation using fixed-step integration algorithms, the solution is to use different integration step sizes for different subsystems, a technique known as multi-rate simulation [7]. In order to ensure accurate and stable performance from a multi-rate real-time simulation it is important not only to partition the system and choose step-sizes with care, but also to manage the exchange of information between partitions carefully. For example, it may be necessary to perform some form of averaging of outputs from the faster components to provide suitable inputs to the slower partitions, and it may be advisable to estimate intermediate values of the outputs from the slower partitions that are used as inputs to faster partitions.

Consider a simulation that consists of two segments running with two different frame rates, one fast, one slow. Some kind of averaging process may be necessary in transferring data from the fast segment to the slow segment. Otherwise the value of a rapidly changing variable that is sampled only at the times that data is required by the slow segment can be unpredictable and can cause aliasing because the sampling frequency represented by the step size of the slower segment is too low to handle the higher frequency components of the output of the faster segment. This problem can sometimes be avoided if the partition boundary can be drawn at a point at which the outputs of the fast segment are known to be slowly varying, for example, at the output of a low-pass filter.

In the opposite direction data from the slow segment can be assumed to pass through a zero-order hold. In other words it is held constant until updated at the end of the frame. Alternatively, a first-order hold can be used. This involves the estimation of the value of the slow segment output at the fast segment intervals that occur during the slow frame in a similar way to the data transfer process used to communicate between modules. If the derivative of the relevant output variable is calculated in the slow segment, it can also be passed to the fast segment and used to make a linear prediction of the intermediate values. Otherwise the derivative can be approximated using the two past values of the variable at the end of successive frames. A more flexible approach is to use fractional-order hold which involves a linear combination of the zero and first-order values.

2.2 Distributed Multi-Rate Simulation

In a real-time multi-rate simulation that involves HSRT elements, the non-HSRT i.e. longer frame-time parts of the simulation will be executed on one or more separate platforms. The obvious choice for these parts of the simulation is a computer, or computers, running some version of real-time Linux, which is widely used for real-time simulations with frame times of about 20 μS or longer. Some companies offer real-time

Linux-based computer systems, usually with added simulation support software and digital and analog I/O for hardware-in-the-loop configurations, which are designed specifically for implementing real-time simulations. Two of the best known providers of these systems are Applied Dynamics inc. of Ann Arbor, Michigan and OPAL-RT of Montreal, Quebec, Canada. The project was provided with a RT-Lab system from OPAL-RT using ONR funding. This consists of a dual-Opteron configuration which executes simulation code under the Red Hawk Linux operating system. Models can be developed using Simulink blocks. Digital and analog I/O is provided. A field-programmable gate array (FPGA) can be user programmed either for custom I/O or to support limited high-speed execution of parts of the simulation. The RT-Lab system provides an opportunity for distributed, multi-rate, real-time simulation using a combination of HSRT and Linux platforms.

The main thrust of the project was to investigate the use of multi-rate distributed real-time simulation and its application to power electronic systems. This study was to consist of both experimental and theoretical investigations, including the extension of previously developed analysis to multi-rate simulation. The policy of integrating the simulations into a VTB environment was to be continued. Attention was also to be paid to the automation of the model development process.

The following sections discuss the work done and results obtained for different aspects of the research. These are: high-speed processors and HSRT implementation (Section 3); bi-rate analysis (4); multi-party, multi-rate simulation (5); and distributed multi-rate simulation (6). Conference and workshop participation is listed in Section 7; Section 8 contains a summary and conclusions. Figures are collected together in Section 9 and references appear in Section 10.

3.0 HIGH-SPEED PROCESSORS AND HSRT IMPLEMENTATION

3.1 Performance with Digital Signal Processors

At the start of the period of performance, the minimum frame time to execute the 6-pulse modularized benchmark simulation using a board containing an array of 4*TS201 digital signal processors was 3.8 μS. There was still some margin for improvement by fine tuning of the coding. A detailed study of the timing of events in each processor was undertaken and this was helpful in determining where improvements could be made. One source of improvement was to reassign tasks between processors to equalize the loading on the four processors. Further improvements in coding which shortened execution times for some data transfers and simulation code were also introduced. A frame time of 2.02 μS resulted from these efforts. Figure 3.1 provides a breakdown of the timing for the four processors.

At this point the TS201 board failed. Extended discussion with the manufacturers concluded that an attempt to repair the board would be costly and with limited hopes of success. A detailed assessment of the performance of the TS201 led to the conclusion that very little further improvement in frame time was possible and that it was not worth the risk associated with attempting repair, or the additional cost of acquiring a replacement board. Discussions with the manufacturers of both the processor chips and the boards on which they are installed also indicated that there were no new products in the pipeline that were likely to offer improved performance for high-speed real-time simulation applications. As a result of these investigations a decision was made to abandon the digital signal processor as a source of further improvements in performance, and to investigate the use of the field-programmable gate array (FPGA). This topic is the subject of Section 3.3. The next section deals with distributed, multi-rate, real-time simulation.

3.2 Distributed Multi-Rate Real-Time Simulation

In parallel with the work on the TS201, an investigation of the newly acquired RT-Lab system was undertaken. This system was seen as a possible key component in a distributed multi-rate real-time simulation in which high-speed simulation of power electronic components implemented on DSP or other high-speed platforms is combined with simulations of the rest of the system running with longer frame times on a more conventional real-time Linux platform.

The RT-Lab system provides two dual-core Opteron processors operating under the Red Hawk real-time Linux operating system. It also provides a field-programmable gate array (FPGA) which can be used either to customize an interface to external equipment, or to provide high-speed simulation of part of the system. Simulation programs can be written in C or, more conveniently in most cases, programmed using Simulink. The 6-pulse benchmark was first implemented on the RT-Lab using Simulink S-fuctions. An effort was then initiated to code at least part of the simulation on the available FPGA. It proved possible to set up a version of the 6-pulse benchmark in which one converter was implemented on the FPGA and the rest of the system on the dual Opterons. Preliminary measurements of achievable frame times indicated a minimum in the range of 10 to 20 microseconds. In contrast to the situation with the DSP boards, the RT-Lab does not

provide a hard real-time solution because interrupts from the real-time Linux operating system are somewhat unpredictable. This means that frame-time measurements during a simulation reveal considerable variations in individual frame times.

The experience with the FPGA component of the RT-Lab suggested that a more powerful FPGA might offer an alternative approach to high-speed real-time simulation. As a result of this limited success attention turned to implementation of the entire 6-pulse benchmark on a larger, stand-alone FPGA.

3.3 High-Speed Real-Time Simulation with the FPGA

A Xilinx Virtex 5 development board and software were purchased for this purpose and a full implementation of the 6-pulse simulation was implemented on a single Virtex 5 FPGA. This has executed with a frame-time of 400 nanoseconds which represents an improvement of 80% over the best performance achieved with the DSPs.

Programming of the FPGA uses a special Simulink blockset. This provides low-level FPGA programming since it represents the various functional components of the FPGA as individual blocks. In order to program a mathematical model based on differential equations it is necessary first to combine the differential equations with the chosen integration algorithm to produce a set of difference equations. The math models used in this research consist of linear ordinary differential equations combined with switching logic for the switches, sine/triangle waveform comparison to establish the switch timing, and digital control algorithms for the PWM feedback controllers. The arithmetic operations in the model consist mainly of repetitive multiply and add operations, which map well to the "DSP slices" in the FPGA.

The Virtex 5 is one of the larger currently available devices of its type. Even so, programming of the 6-pulse benchmark can be achieved only by sharing of components which perform multiple multiply and add operations sequentially. This is done by multiplexing several pairs of inputs (up to 4 in this case) to the input of the arithmetic unit and selecting them sequentially and storing intermediate results before finally combining them in a sum of products. There is clearly a trade-off available between size of problem and speed. This offers promise of being able to simulate significantly larger models on the same device at a cost of extending frame times to, say, one or two microseconds, which may be sufficiently short for many practical applications. Alternatively, if even shorter frame times are demanded, or if larger systems need to be simulated with frame times in the 400 nanosecond range, there is the option of combining a simulation on more than one FPGA. This possibility will be investigated as part of the future research activity.

At the conclusion of this project, output from the FPGA simulation was restricted to oscilloscope displays of signals that allowed basic measurements to be made of simulation performance. Future research will address the development of high-speed connections to selected computing platforms and interfacing the FPGA simulations to the VTB and other user interfaces.

It should be pointed out that the process of implementing a high-speed real-time simulation on one or more FPGAs is currently very labor intensive and time consuming. The claim that simulations are programmable using Simulink is misleading. It is not possible at this stage to automatically convert normal Simulink blocks into FPGA implementations. The programmer is restricted to the use of the special FPGA blockset which does facilitate the process, but which only maps low-level FPGA function blocks directly into a Simulink flowchart. This is a tedious process and it is only practicable if the math model of the simuland is first converted from differential equation form into difference equations. This is done by combining the differential equations with the selected integration algorithm, a process which is also a necessary stage in programming DSP simulations. The other parts of the math model (switching logic, PWM control functions) are more readily converted into FPGA functions.

Some progress has been made in automating the differential equation to difference equation conversion using Mathematica. This development is described in Section 5. First, the continuing work on stability analysis is described in the next section.

4.0 STABILITY ANALYSIS

4.1 Summary of Previous Work

Earlier research addressed the analysis of the stability of different candidate numerical integration techniques [1]. This led to the selection of state transition methods as the best choice for the type of model of interest in this research. These studies included investigations of the effect of modularizing the models and of different methods of communicating data between modules. The results of the analysis was validated in each case by performing simulations using a variety of methods including direct evaluations of the difference equations, trial simulation runs using the developed code for the real-time simulation, and the use of simulation packages such as Matlab and Spice. The primary purpose of these calculations was to confirm that the stability conditions predicted by the theoretical analysis were supported by direct simulations. A valuable by-product of this process is that it provides valuable information that can be used to verify the correctness of the developed program. Whenever discrepancies were revealed between the various solution methods, further study investigated the source of the problem, which was often traced to errors in the real-time simulation code. This proved to be a significant time-saving technique that high-lighted the presence of coding errors before the code was converted for running on the DSPs.

4.2 Bi-Rate Simulation Analysis

As mentioned earlier, a major thrust of this project was to investigate the use of multi-rate simulation techniques for integrating both high-speed and normal-speed components into a real-time simulation of a complete system. Multi-rate simulation involves the partitioning of the system to be simulated into modules or segments with different dynamic characteristics. Each segment is simulated using a frame-time appropriate to the dynamics of that segment. With the restriction that the frame-time ratios must be integers, this gives the programmer freedom to select the integration step-size that is considered most suitable for a particular subsystem. It is also possible to use different integration algorithms for different segments, and this was the case in some of the situations described later in this section.

The use of multi-rate techniques meant that the stability analysis that was used in the earlier work required extension to consider the multi-rate case. The research performed so far has been limited to bi-rate simulation. This was done as a starting point for a more general multi-rate analysis. However, analysis of simulations using more than two rates is much more difficult than the bi-rate case and no satisfactory method of completing it has so far been identified.

The bi-rate analysis that has been completed so far has included analysis of the Euler, state transition (3.2 method), and implicit trapezoidal methods [6]. The effect of different methods of data transfer (zero-, first- and fractional-order hold) with different integration algorithms has also been studied. The results of these analysis have confirmed that, for the piecewise continuous linear systems that form the basis of the power electronic models on which this research is focused, the state-transition 3,2 method offers the best performance in terms of accuracy, stability, and computational cost. The bi-rate stability

analysis provides a good basis for determining the ranges of parameter values that can be relied upon to guarantee stable simulations. This can guide the choice of algorithm. The state transition approach provides a class of methods of which (3,2) is one of the simplest while providing excellent performance in most cases. When (3,2) proves unsatisfactory a more complex method involving more terms can be a better choice. It is a feature of the state transition approach that the complete infinite series solution is always stable if the original differential equations are stable. There is, therefore, a trade-off between stability and speed that can be exploited in difficult cases.

4.3 Comparison of Implicit Trapezoidal with State Transition Methods

For many years the numerical integration algorithm of choice for power-system applications has been the implicit trapezoidal method. This method is guaranteed to be stable if the model differential equations are stable and in common with other implicit methods it has good accuracy characteristics. It can, however, require significantly more computational effort than the widely used explicit methods, particularly when solving non-linear differential equations. This is because of the implicit form of its calculation. At each step it is necessary to solve a set of simultaneous algebraic equations to resolve the implicit relationship. If the differential equations are non-linear then an iterative process is necessary to achieve this. Multiple iterations at each step produce three adverse consequences for high-speed, real-time simulation: first is the extra computation time caused by the iterative solution, second is the uncertainty about the computation time in each step caused by the variable number of iterations that may be necessary, and third is that future values of external inputs are required. The first two problems can be addressed when the differential equations are linear, because the resulting simultaneous equations are also linear and can be solved using a time-deterministic process such as Gaussian elimination or triangular decomposition. The problem of future values of inputs can only be resolved by using some form of prediction of future values, and this compromises the stability characteristics as well as the accuracy of the method. Since the simulation equations in this research are linear, it was decided that the implicit trapezoidal method should not be rejected without performing a study of how it compares to the state-transition methods which have been adopted as standard in the project.

An analytic study was undertaken to compare implicit trapezoidal and both the full and truncated versions of the state-transition method. Stability criteria were generated and two test examples used to compare performance. The conclusions of this study were as follows (ITM=implicit trapezoidal method; FST=full-series state-transition; TST= truncated-series state-transition).

Stability characteristics
Both the ITM and FST methods are stable if the system ODEs are stable.
The FST method is more stable than ITM for large steps.
The stability of TST methods depends on series length and step size.
Computational complexity
The ITM method requires solution of an m*m set of linear equations and future inputs
The FST method requires evaluation of an infinite series m*m matrix
The SST method requires fewer multiplications than FST or ITM

The conclusion is, that for the type of mathematical model of interest in this project, the state-transition methods offer a better choice than implicit trapezoidal. The choice of which state-transition method to use involves a trade-off between speed, accuracy and stability, but the (3,2) method has been found to work well in most cases.

4.4 Frequency Domain Methods for Step-Size Selection

The selection of a satisfactory value of integration step-size is one of the key decisions in the design of a simulation involving differential-equation-based models. The traditional approach is to use a combination of knowledge of the system dynamics and a trial and error approach in which time-domain solutions for different step-lengths are compared to determine whether reducing the step-length has produced a significant change in the time-histories generated by the simulation. If a step-length reduction (say by half) produces no significant change in the time histories, the assumption is that the error produced by either method has fallen below the critical level and that the longer of the two step-lengths involved in the comparison is acceptable. While the reasoning behind this approach is simple, its effective implementation often is not. For example, the test for a particular subset of variables and parameter values doesn't necessarily apply to all variables or the full range of parameters. An exhaustive investigation for all variables and a comprehensive coverage of parameter space is usually not feasible. A lot of system knowledge and judgment is involved in making the best choice.

As an alternative to these time-domain methods a study was made of possible frequency-domain techniques based on a similar approach [8]. This involves comparing frequency spectra of the simulation outputs for different step-lengths in place of the time-domain histories that are normally used. The same reservations apply to this approach as to the time-domain method, but the possibility that the frequency-domain data gives a clearer perspective of some aspect of the system behavior may provide additional insights that facilitate the selection of the time-step.

On this basis, an investigation of frequency-domain studies was undertaken using the 6-pulse benchmark as the object simulation. The approach used was to generate power spectral density (PSD) functions of selected simulation output variables for different step-lengths as a basis for comparison. The amplitudes of different frequency components in the spectra could then be studied to gain insight into the effect of changes to the step-length. Since the PSD is generated from a sample of finite length it is necessary to window the data carefully, that is to pre-process it before the application of a fast Fourier transform (FFT) that produces the PSD. A thorough study was made of the available windowing approaches and the Blackman-Harris window was selected as the most suitable for this purpose [9]. The data was also used to generate a measure of total harmonic distortion (THD) of the waveform. These processes were subsequently refined to produce PSD and THD tools integrated within both VTB 2003 and VTB Pro.

The use of these tools to assist in step-size selection was investigated. The conclusion was that they added valuable information about the correctness of the behavior of the simulation when viewed by a domain expert. In particular the emergence of certain

frequency components, particularly at lower frequencies, as step-length was increased provides useful clues in choosing integration step-length.

5.0 MULTI-PARTY MULTI-RATE SIMULATION

5.1 Real-Time Multi-Rate Simulation

A major thrust of the project was to investigate ways in which high-speed real-time simulations can be used in simulations of complete power systems. Special techniques are necessary to achieve the short frame times of a few microseconds or less necessary for accurate real-time simulations of modern power electronic subsystems. These techniques are not, however, either required or appropriate for simulations of other parts of the system, such as motors, generators, mechanical components, thermal calculations etc. This suggests a distributed simulation that combines special high-speed components with more conventional real-time simulation platforms. This approach is inherently multi-rate, since the different parts of the simulation will use frame-rates appropriate to the dynamics of the relevant subsystem.

The multi-rate approach raises questions of accuracy and stability for the complete simulation. It is well established that problems can arise when a combination of stable subsystem simulations produces an unstable simulation of the combined system. This was the motivation for the bi-rate stability studies described in Section 4.

In this section we take the discussion a stage further. It is not uncommon for the different parts of a total system simulation to be developed by different parties who have expertise in the relevant discipline (electrical, mechanical, structural, thermal, chemical etc.) using different simulation tools and techniques. This further complicates the process, particularly for the integration of the various subsystem simulations into a single, coherent, stable, accurate simulation. This process is designated here as multi-party, multi-rate simulation. It is an important research topic from both the real-time and non-real-time perspective. In order to identify the important issues and challenges of multi-party multi-rate simulation, a benchmark simulation was defined to be developed by three teams from different organizations.

5.2 Multi-Party Multi-Rate Benchmark

In order to investigate the problems of multi-party multi-rate simulation a consortium of teams from three universities (California State University, Chico, University of South Carolina, and Glasgow University) collaborated on the development of a simulation of an unmanned underwater vehicle. The system consists of three main subsystems, the power supply consisting of a battery and inverter, the electric drive, and the vessel. CSU, Chico developed the inverter model with controller, USC was responsible for the battery and the electric drive, and Glasgow University developed the model of the vessel. USC also developed a multi-rate solver for the VTB.

5.3 UUV System Simulation

The UUV system is illustrated in Figure 1. The d.c. power source simulation is taken from the VTB simulation library. The discharge characteristics of each battery cell are simulated, and cells are connected in a series-parallel configuration to achieve the desired voltage and capacity.

The d.c. power source is connected to the six-pulse, three-phase d.c. to a.c. PWM converter. The converter contains a six-switch network that produces a synthesized, three-phase, variable-frequency a.c. waveform. The a.c. output from the converter is filtered to remove high-frequency harmonics and is then passed to the induction motor via a natural coupling.

The converter switches are operated by on-off commands from the PWM switch controller. A feedback controller is used to control the frequency of the converter output, which ultimately controls the propeller RPM of the UUV platform. The controller also has the capability to use a PI scheme to control output current or power supplied to the motor drive. The controller determines switch timing by the comparison of sine waves at the output frequency and five-kilohertz triangular waves. Their relative amplitudes are adjusted by the feedback control and switching occurs when the sine and triangular waves intersect. The phase of the a.c. output may also be controlled.

The converter simulation is implemented in C++ as a native VTB model. The d.c. input connection and the a.c. output connections are natural couplings. The connections to the controllers are signal couplings. Controller models contain only logical elements and do not have any differential equations. The converter simulation uses trapezoidal integration for the filter components, and Euler integration for the input d.c. filter capacitor. The sine and triangular waveforms are generated by table lookup, scanning the tables at a rate determined by the desired frequency. Linear interpolation is used, with enough table entries to have a maximum of a one-bit error after interpolation.

The converter is used to supply a.c. power to the induction motor. The motor simulation is implemented in C++ as a native VTB simulation, and uses trapezoidal integration. The motor applies torque to the propeller shaft and rotates the propeller at a speed determined by the input a.c. frequency, driving the UUV forward. The mechanical connection between the motor simulation and the ship simulation is a signal connection. The motor model outputs an RPM and the combined ship and propeller model returns the torque on the shaft. The input a.c. power to the motor uses a natural connection.

The ship has a six degree of freedom simulation, originally implemented in Matlab and uses fourth-order Runge Kutta integration. The fin deflections are input via the user interface and propeller shaft RPM as a signal input. The Matlab simulation was translated to C++ and implemented as a native VTB model, although the original model may still be used via the VTB/Matlab interface. Typical simulation frame rates used in this simulation are:

- Battery, ship, 3D graphics*: 100 mS
- Converter, switch controller: 2 μS
- Feedback controller: .8 mS
- Motor: 100 μS

5.3.1 Multi-Rate Implementation

There are two implementations of the UUV system using two different techniques for providing multi-rate capability. One implementation uses a distributed multi-rate solver

that is provided as an integral part of the VTB environment. To use this multi-rate VTB solver, one simply partitions the system with each partition containing models to run at the same time step. Connections between the partitions terminate in special VTB ports. Each partition is distributed to a separately running copy of the VTB, connected by an Ethernet network. The partitions can reside in a single computer or in distributed computers. The VTB distributed solver was not available during the initial stages of the UUV implementation, so a simple, special-purpose, multi-rate solver was created to facilitate early development and testing.

In this simple solver a collection of models to be run at different rates were bundled together as a single "super-model" from the perspective of the VTB. The VTB ran at the rate of the slowest model. Other models had to run at a integral divisor of the VTB step size. Each VTB time-step the model bundle was called by the VTB, and an internal scheduler would call the internal models at an appropriate rate, returning to the VTB when the time had elapsed that corresponded to the VTB step size. Natural couplings between internal models were handled by a special purpose internal solver.

The advantage of this simple solver was a slightly lower overhead compared to that of the general-purpose VTB distributed solver and because of its early availability. The disadvantage was that results could only be displayed on the user interface at the VTB time step of the slowest model. Results from fast electrical circuits can not be accurately displayed if there is a high multi-rate factor.

5.3.2 VTB Component Connections

Conventional integration algorithms use the rates of change of the state variables at time, t_n, to advance the solution by a time step, dt, using the chosen integration algorithm to produce the values of the states at t_{n+1}. These can then be substituted in the differential equations to update the rates of change at t_{n+1} and the process is repeated. This type of output from a model is called a signal connection in the terminology of the VTB. Outputs are numbers, and inherent in their calculation is frequently a knowledge of the load created by external connections. An example output would be a current computed for a connection with a given input voltage.

The VTB, on the other hand, allows the interconnection of multiple arbitrary system components without having to change the coding of the individual components. To achieve this, "natural" couplings are used between components. A natural coupling is neither an input nor an output, but it is a connection where the values at each end of the connection must be consistent. For example, in an electrical circuit, the voltage at any connection point is the same for all connected devices, and the sum of all the currents at that point must be zero. To accomplish this, a component does not output a current given the voltage at a connection point, but outputs the equation that relates what the current would be for any voltage. An external VTB solver then computes what the voltage would have to be so that the sum of the currents from all connected models would be zero. The equation has the following form:

$$\mathbf{I} = \mathbf{G}*\mathbf{V} - \mathbf{b} \qquad (1)$$

Each model is called at time t to output the value of G and b for time $t+1$.

Methods for computing **G** and **b** typical involve algebraic manipulation of the difference equations that result from applying an implicit integration method to the model's differential equations.

The varied simulations found in the UUV system have all been developed at separate locations for different simulation environments and written in different languages. Modules exist in the VTB to interface to many other simulation environments, but VTB natural couplings were not directly compatible with traditional signal inputs and outputs. A screen shot of the VXE 3D graphics representation of the UUV during simulation is shown in Figure 2.

In order to integrate these models into a single system, several approaches were used. For example, the ship model was developed in Matlab and used signal connections. The VTB/Matlab interface solved the language compatibility issue, but the mechanical connections for torque and RPM were not directly compatible with the natural connections for the VTB motor models. In this case, the original VTB model equations for a simple motor model were re-implemented to have signal inputs and outputs for the mechanical connections.

Another approach for connecting signal to natural ports was to use the VTB models for constant current and constant voltage sources to generate natural outputs from signal outputs. This worked in some cases, but tended to add instability to the system.

The choice of a UUV model as a benchmark for investigating multi-party, multi-rate simulation has proved to be a good one. The model, although quite simple, has generated many of the problems that one might expect in larger more realistic applications. There is a lot of potential for developing the model. The Glasgow team has experience of designing and constructing functioning physical model UUVs and it would be feasible to use simulation to design a physical model which could then be used to validate the simulation.

6.0 DISTRIBUTED MULTI-RATE SIMULATION

6.1 Real-Time Linux Systems

A key component of a distributed multi-rate real-time simulation system is the computing platform that is used to implement the slower subsystem simulations that are interfaced to the high-speed simulation elements. A number of commercial systems are available, for example, the RT-Lab from Opal-RT, the rTX from Applied Dynamics and the D-Space system. During the period of performance for this project the research had obtained a RT-Lab system. Discussions were proceeding on the procurement of an rTX system but this was not yet available for the research.

6.1.1 RT-Lab System

After some initial problems with the RT-Lab system caused by a hardware fault on the delivered system combined with incompatible software components, the system installation was completed. The RT-Lab provides two dual-core Opteron processors (four processors in all) as the main simulation platform. Code can be written in C or Simulink and runs under the Red Hawk real-time Linux operating system. FPGA-based interfaces are provided for both digital and analog I/O. An additional FPGA is provided that can be used either to provide a custom interface or to implement high-speed real-time simulation components that can be combined with the simulations running on the dual Opterons.

Initial research using the RT-Lab system had two main goals. First was to investigate the implementation of power electronic models on the system, second to develop a high-speed interface between the RT-Lab and the TS201 digital signal processor board. The first step was to run the 6-pulse benchmark on the RT-Lab using Simulink S-functions and this was completed. The interfacing problem was approached from both ends with FPGAs at both ends of the link successfully programmed for simple I/O. This was the stage of development reached when the DSP board failed and this effort was suspended at this point.

Attention was then directed towards implementing the 6-pulse simulation on the RT-Lab FPGA. It soon became clear that, given the limited capacity of the Xilinx Virtex 2 device provided for this purpose, it would not be possible to implement the complete system in this way. Instead one converter was programmed on the FPGA and the rest of the system ran on the dual Opterons using Simulink S-functions. This experience was very valuable in introducing the team to the use of a Simulink FPGA blockset and to the techniques and trade-offs involved in programming difference equations for FPGA execution.

Given the failure of the DSP and the experience gained with the FPGA on the RT-lab system, a decision was made to obtain a development system containing a more powerful FPGA with a view to implementing the complete 6-pulse benchmark. A Xilinx Virtex 5 development board was ordered for this purpose and was undergoing initial testing and evaluation at the end of the period of performance for this project.

7.0 CONFERENCE AND WORKSHOP PARTICIPATION

7.1 Papers and Conference Participation

Papers have been presented at the SCSC (Summer Computer Simulation Conferences) in 2005 (Philadephia PA), 2006 SCSC (Calgary, Alberta, Canada), and 2007 (San Diego CA) at the 2007 WSC in San Diego CA and at the 2005 and 2007 Electric Ship Technology Symposia. A paper was also presented at a European Space Agency Symposium in Noordwijk, Netherlands in September 2006. Dr. Crosbie presented the Keynote at the 1st Asian Modeling Symposium in March 2007 in Phuket, Thailand. Panel sessions on relevant topics have become an annual feature of the Huntsville Simulation Conference in Huntsville, Alabama with strong participation from academic, industry and government representatives. Members of the research team presented a paper at a Conference on the Science of Design at Humboldt State University, Arcata CA.

7.2 Workshop Participation

Members of the Chico team have attended and participated in VTB Workshops and Conferences organized by the University of South Carolina. The Chico team helped to organize and participated in international research workshops at the University of Cambridge (2005) and at Glasgow University (2006, 2007).

7.3 Collaboration with Research Partners

During the project an international collaboration was established between CSU, Chico, the University of South Carolina, Glasgow University, and ISIM International Simulation, UK developers of the ESL simulation language. Chico, South Carolina and Glasgow collaborated on the development of the multi-rate benchmark simulation of a UUV; ISIM provided ESL licenses, instruction in the use of ESL, and modifications to ESL to support the research. ISIM also collaborated with USC and CSU, Chico on the integration of ESL with the VTB.

The collaboration with OPAL-RT continued with support from OPAL for the use of their RT-Lab system.

Discussions were initiated with Applied Dynamics Inc. aimed at obtaining a loan of an ADI rTX system to support the research on distributed, multi-rate, real-time simulation. Discussions were also held with Dr. Robert Howe of ADI and Professor Emeritus at the University of Michigan about real-time numerical integration algorithms. Dr..Howe is an internationally acknowledged expert on real-time simulation and he provided a copy of his notes for the use of the team. This has initiated a review of the algorithms developed by Dr. Howe to evaluate their potential for use in the research.

8.0 SUMMARY AND CONCLUSIONS

The initial goal of the research was to develop techniques capable of producing real-time simulations of power-electronic systems, using low-cost off-the-shelf components, with frame times of 10 microseconds. This was subsequently reduced to 2 microseconds. This goal has been effectively met using small arrays of digital signal processors. Attention has now been directed towards the use of FPGAs to achieve even shorter frame times and this approach is showing great promise.

Progress has also been made towards establishing a basis for distributed, multi-rate, real-time simulation of complete power systems. Bi-rate analysis of the numerical algorithms has been made, and a real-time Linux system, RT-Lab obtained for experimental study.

A benchmark for multi-party multi-rate simulation based on an unmanned underwater vehicle has been developed in collaboration with research partners, and an initial implementation has been completed. Further work is required to complete this study.

Results of the research have been disseminated at several conferences, workshops, and symposia, and the research team is gaining international recognition for its contributions.

9.0 FIGURES

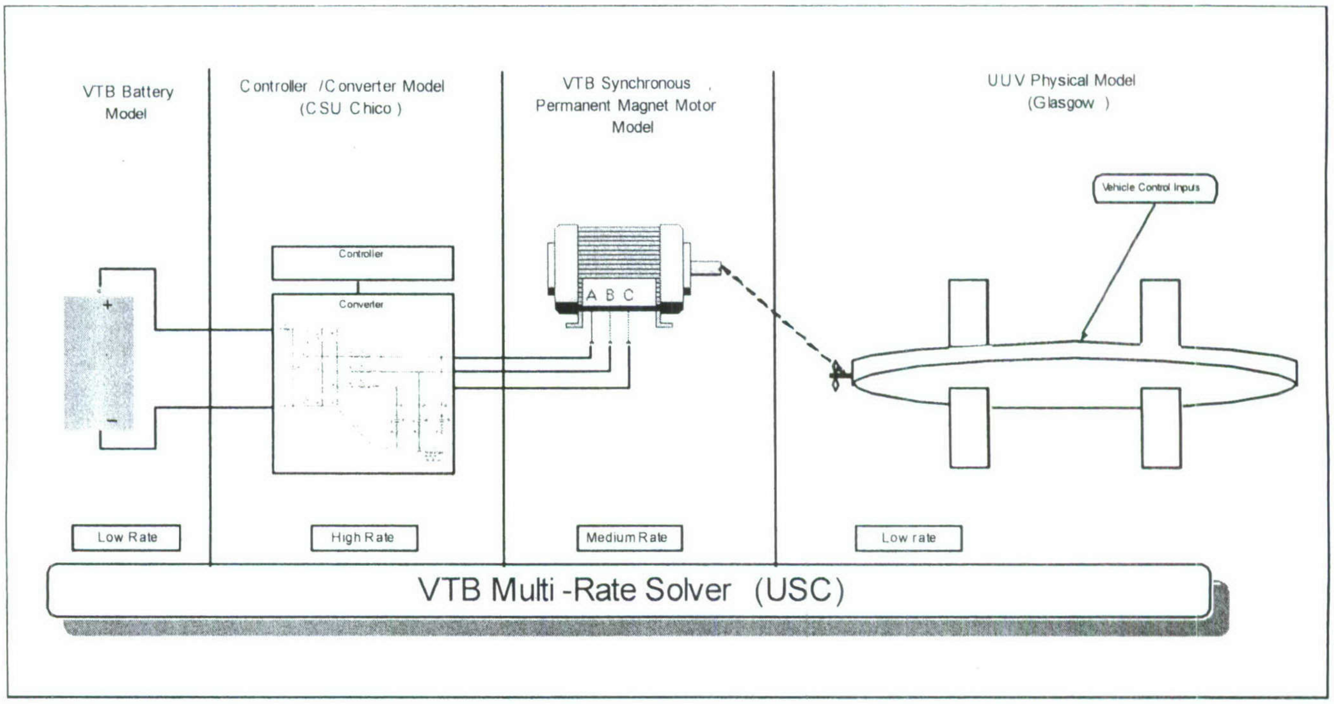

Figure 1: Simplified representation of UUV model.
(Details of vessel input and output variables and controls are omitted

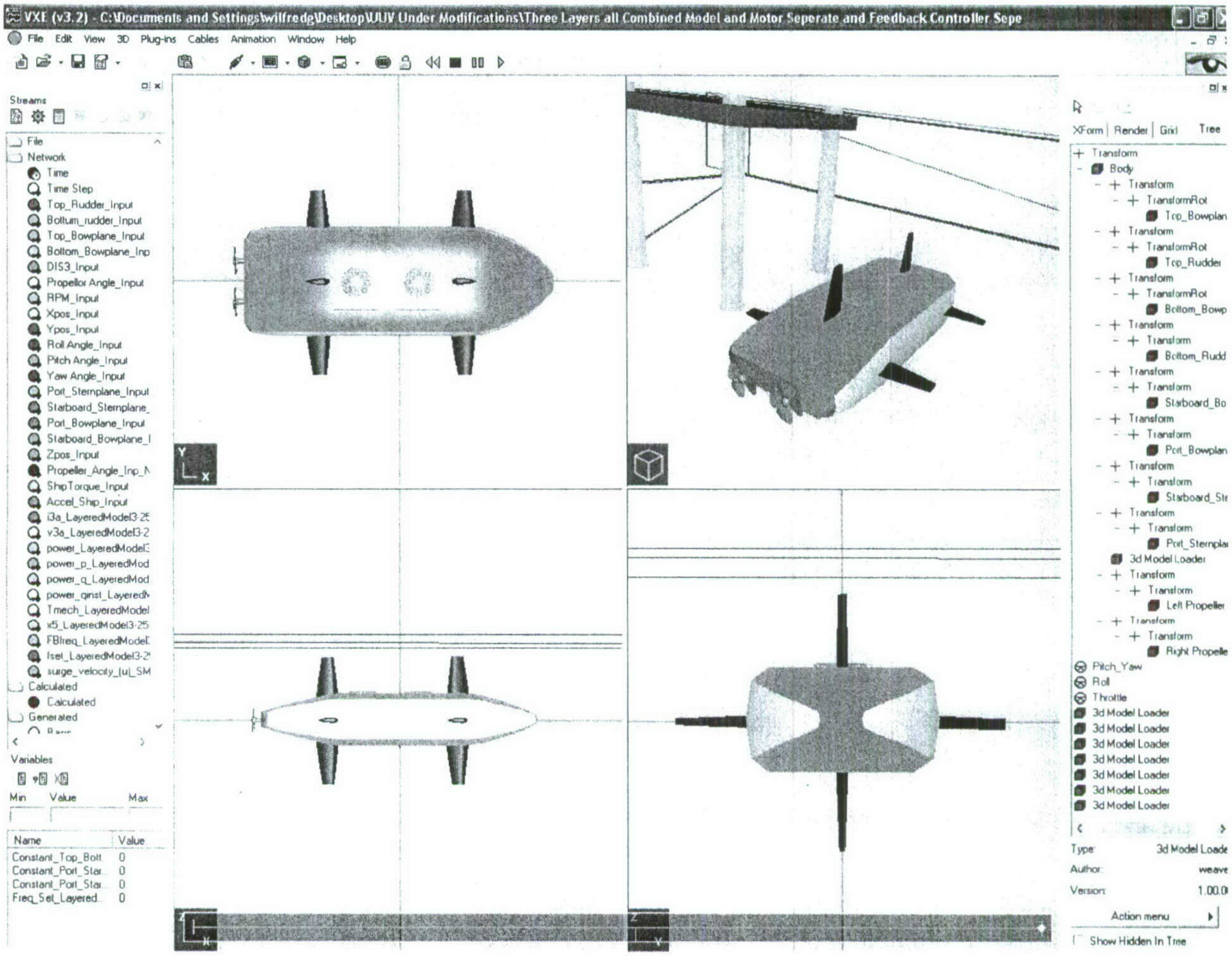

Figure 2: VXE Screen-Shot of UUV Simulation

10.0 REFERENCES

[1] Crosbie, R.E., J.J. Zenor, R. Bednar, D. Word, N. Hingorani, and T. Ericsen, *High-Speed, Scalable, Real-Time Simulation Using DSP Arrays,* Presented at IEEE/ACM Parallel and Distributed Simulation Workshop, Kufstein, Austria, 2004.

[2] Crosbie, R.E., J.J. Zenor, D. Word and N.G. Hingorani "Fast Real-Time DSP Simulations for On-Line Testing of Hardware and Software" *Modeling & Simulation*, Vol 3, No 4, Oct-Dec 2004, SCS San Diego

[3] R.E. Crosbie, J.J. Zenor, R. Bednar, D. Word and N.G. Hingorani, *PC-DSP Based Expandable, Real-Time Power-System Simulator (RTPS) for Power Electronics,* ONR Award #N00014-01-1-0394, Final Report, California State University, Chico, March 2005

[4] Dougal, R. A., S. Liu, L. Gao, M. Blackwelder, *Virtual Test Bed for Advanced Power Sources,* J. of Power Sources, Vol. 110, No. 2, pp. 285-294, 09/02.

[5] Dougal, R. A., "Design Tools for Electric Ship Systems" in *Proceedings of IEEE Electric Ship Technologies Symposium,* Philadelphia PA, July 2005 pp. 8-11, IEEE.

[6] Bednar, R. and R. E. Crosbie, "Stability of multi-rate simulation algorithms" in *Proceedings of 2007 Summer Simulation Multiconference,* San Diego, CA, July 2007.

[7] Word D., J. J. Zenor, R. Bednar, R. E. Crosbie and N. G. Hingorani "Multi-Rate Real-Time Simulation Techniques" in *Proceedings of 2007 Summer Simulation Multiconference,* San Diego, CA, July 2007.

[8] Bednar, R., R.E. Crosbie, "Applications of Power Spectral and Harmonic Distortion Techniques to Computer Simulation Parameter Selection", *Proceedings of the Summer Computer Simulation Conference,* Society for Modeling and Simulation International, Calgary, Alberta, Canada, July 2006

[9] Harris, Fredric, *On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform*, Proc. of the IEEE, Vol. 66, No. 1, p. 64, January 1978